COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

417 Advanced Graphics

Author:
Yuan Zhu(part1,2), Jinwei Zhang(part3,4)
(CID: 01531935,01540854)

Date: February 24, 2019

1 Generate plots of Fresnel reflectance

The task of first part of this assignment is to generate plots of Fresnel Reflectance for dielectric materials.
Given incident angle θ_i, index of refractions η_i and η_t, refraction angle can be calculated as $\theta_t = arcsin(\frac{\eta_i}{\eta_t} sin\theta_i)$. The corresponding parallel and perpendicular polarized reflectance can be calculated as:

$$R_{\|} = |\frac{\eta_t cos\theta_i - \eta_i cos\theta_t}{\eta_t cos\theta_i + \eta_i cos\theta_t}|^2$$

$$R_{\perp} = |\frac{\eta_i cos\theta_i - \eta_t cos\theta_t}{\eta_i cos\theta_i + \eta_t cos\theta_t}|^2$$

The unpolarized reflectance is then be $F_r = \frac{R_{\|}+R_{\perp}}{2}$.

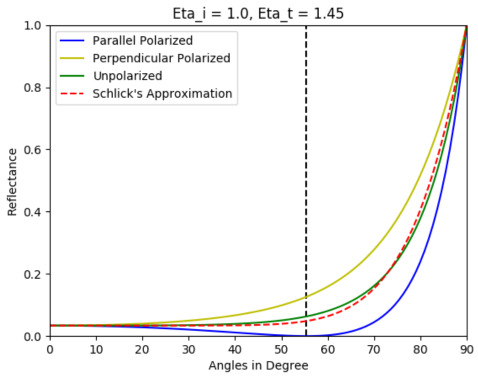


Figure 1: Fresnel and Schlick's Apprx. of η_i=1.0 and η_t = 1.45

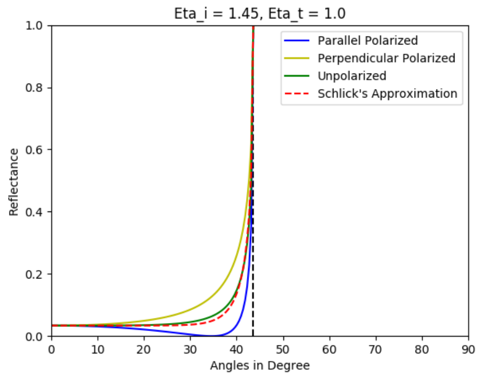


Figure 2: Fresnel and Schlick's Apprx. of η_i=1.45 and η_t = 1.0

As is shown in Figure 1, the Parallel and Perpendicular Polarized reflectance according to incident angle are plotted as blue and yellow solid curves and the unpolarized reflectance is plotted as green solid curves. For index of refractions in Figure 1, the black dotted line represents Brewster's angle, where $R_{\|} = 0$, which has a value of $\theta_B = arctan\frac{\eta_t}{\eta_i} = 55.41(°)$. For Figure 2, the black dotted line is Critical angle, representing the smallest angle that occurs total internal reflection. The value of Critical andgle in Figure 2 is $\theta_C = arcsin\frac{\eta_t}{\eta_i} = 43.60(°)$.
The Schlick's Approximation is shown in red dotted line and calculated by $F_r(cos\theta) = R_0 + (1-R_0)(1-cos\theta)^5$ with R_0 the reflectance at normal incident. According to Figure 1 and 2, it could be concluded that the Schlick's method approximates unpolarized reflectance very well. In addition, the advantage of Schlick's Apprx is to acquire approximated reflactance without estimating index of refractions.

2 Generate MC samples according to an EM

The second part of the assignment is to generate a number of Monte-Carlo samples from lat-long format environment map Grace Cathedral. The generating method is based on Probability Density Function and Cumulative Density Function. For each pixel in lat-long map, the intensity of the pixel is calculated by $Intensity_{ij} = \frac{R+G+B}{3}$. After that, considering the shape of lat-long format, primitive PDF of each pixel should be scaled by its solid angle $\theta(0-\pi)$: $p(x_{ij}) = Intensity_{ij} * sin\theta$.
After the procedure, the 2D PDF of the lat-long EM has been generated. In addition, a 1D PDF that contains sum likelihood of each row is generated for row selection later ($p(r_i) = \sum_1^j p(x_{ij})$). Here, to acquire a proper CDF later, 2D PDF will be normalised as the sum of likelihood of each row equals to 1 ($\sum_1^j p(x_{ij}) = 1$).The 1D row PDF is normalised as well ($\sum_1^i p(r_i) = 1$).
In order to sample from PDF, 1D CDF across rows will be generated next. For each sample, firstly, use uniform random variate $u_i \in [0,1]$ to decide which row to sample. CDF value of row r_m is calculated by $C(r_m) = \sum_1^m p(r_m)$. Apply same operation to rows of 2D PDF to acquire 2D CDF. Given that each likelihood is greater or equal than 0, $C(r_{m+1}) > C(r_m)$ for each m. Therefore, $r_i = C^{-1}(u_i)$ and row number (i) of the sample is acquired. Next, generate another uniform random variate $u_j \in [0,1]$ and use the same method to acquire sample column. Extract row i from 2D CDF and $x_{ij} = C^{-1}(u_j)$. For the assignment, 64, 256 and 1024 samples maps are generated corrspondingly.
After generation of samples, apply scaling and gamma correction to acquire final output. Here the step=6 and gamma=2.2. For clarity, a 5X5 neighbour window around samples are set to blue alone with samples.
For 256 samples, a map only contains sample points and 5X5 windows set to sample RGB values are generated as well.

Figure 3: 64 Samples with Grace Cathedral EM

Figure 4: 256 Samples with Grace Cathedral EM

Figure 5: 256 Samples extracted from Grace Cathedral EM

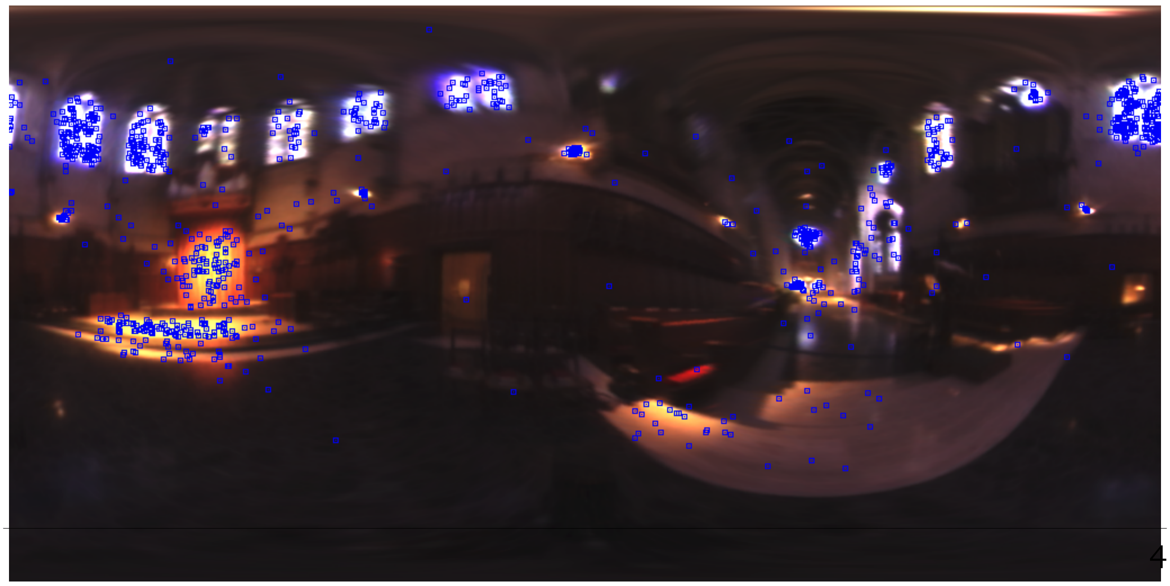


Figure 6: 1024 Samples with Grace Cathedral EM

According to Figure 3 to 6, it can be concluded that comparing to dim pixels, brighter pixels have greater chance to be sampled. Comparing to rows closing to top and ground, rows closing to centre have greater chance to be selected.

3 Generate samples on an EM using Median Cut

In this section, we are given an grace cathedral environment map in lat-long format (.pfm) we need to recursively partition the EM into two halves with equal energy/intensity I = (R+G+B)/3

The basic algorithm is shown as follow:

1 read the image as .pfm format using ***loadPFM()*** function read the height and width.

2 instantiate a intensity array to store the intensity for each pixel. using $\frac{R+G+B}{\sin(\theta)}$ θ represent the latlong map varies from the top to bottom $\theta = 0$ - π.

3 call the median cut partition algorithm recursively to cut the image into equal intensities (described below) and draw white lines to separate each partitions

4 draw a 5*5 box around each centroid of the partition in Blue

5 write out the changed image and apply gama function. In the 64 partitions case also showing only the centroids representing the total RGB radiance within each partition

The median cut algorithm is described as follow:

1 each time pass in the start row index, end row index, start column index, end column index(for calculate the centroid), and iteration number, and the image(for modification)

2 base case is then condition when iteration number is equal to the partisan number(partitions = $2^{\text{partition number}}$). if the base case is met then stop cutting.

3 If the base case is not satisfied then firstly find the longest dimension using the parameter passed, note that this dimension will be cut acrossed. using the intensity array described above to find the middle index along the longest dimension and set the whole (line/column) into white to separate two equal regions.

4 record two centroid of the partitions then call the median cut algorithm two times with two sub regions also modify the parameters respectively.

5 when showing only the centroids representing the total RGB radiance within each partition in the 64 partitions case we also need another array to store the RGB total value corresponding to the centroid array.

6 the algorithm will terminate once the base case is met.

the algorithm performs like a recursive Depth-First-Search algorithm. The result is shown as follow: (256 partitions). One thing to be careful is using a hard copy the original image instead of a shadow copy to calculate the total RGB. Because if we use a modified image (including white line) the total RGB will increase a lot.

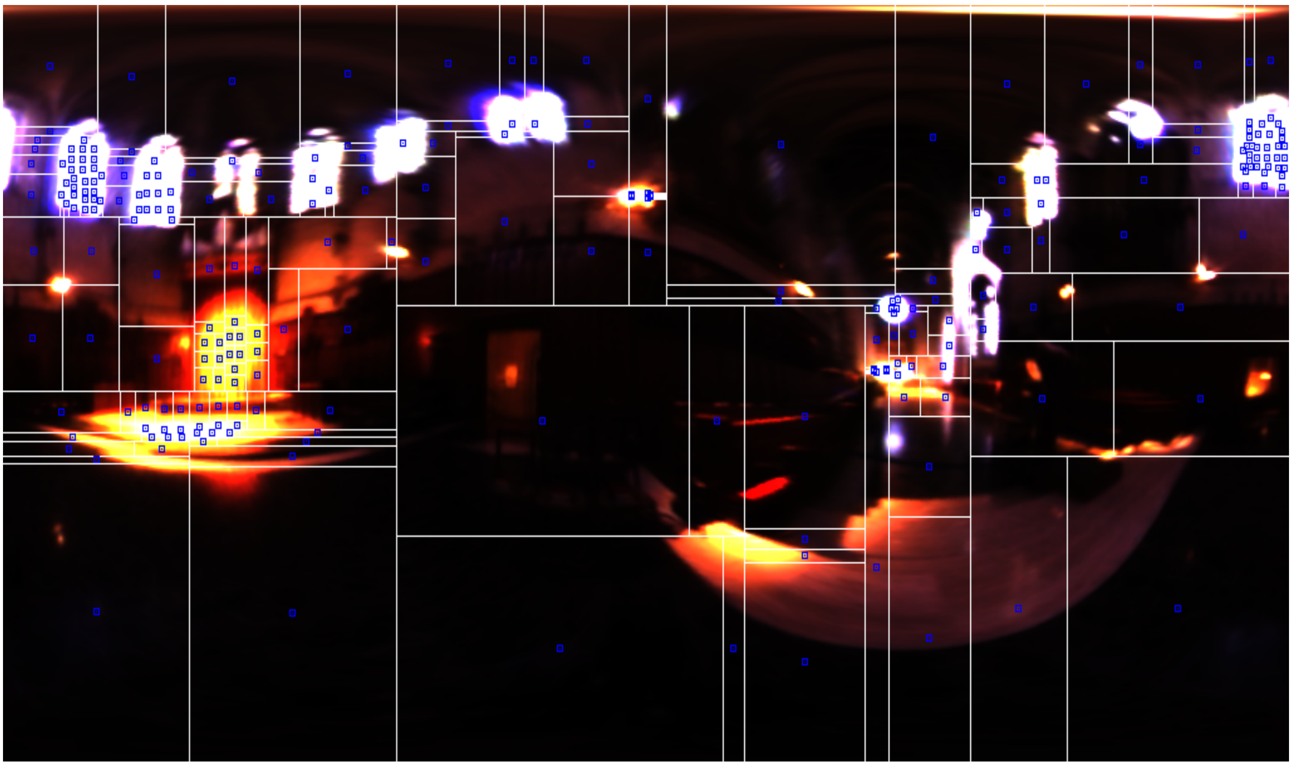


Figure 7: 256 partitions EM

Figure 8: 2 partitions EM

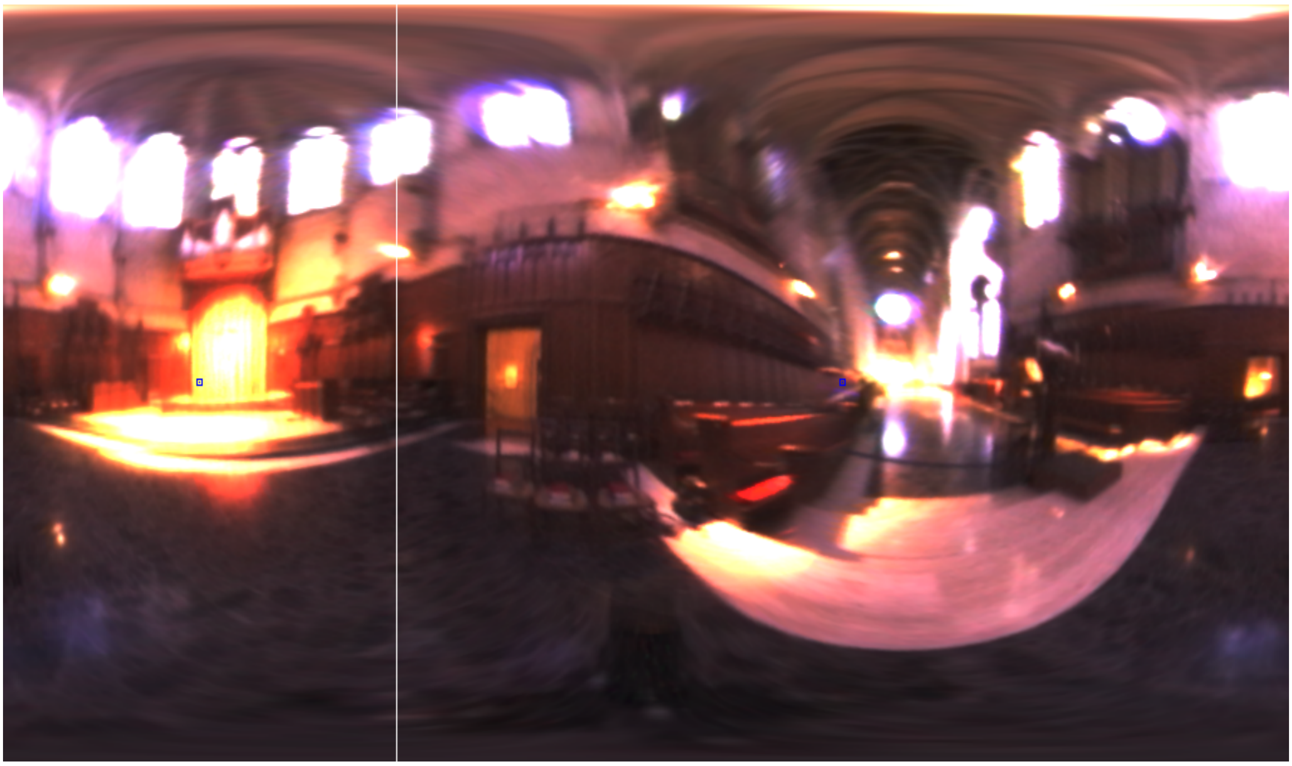


Figure 9: 2 partitions EM after gamma correction gamma=1.8 stop=3

Figure 10: 4 partitions EM

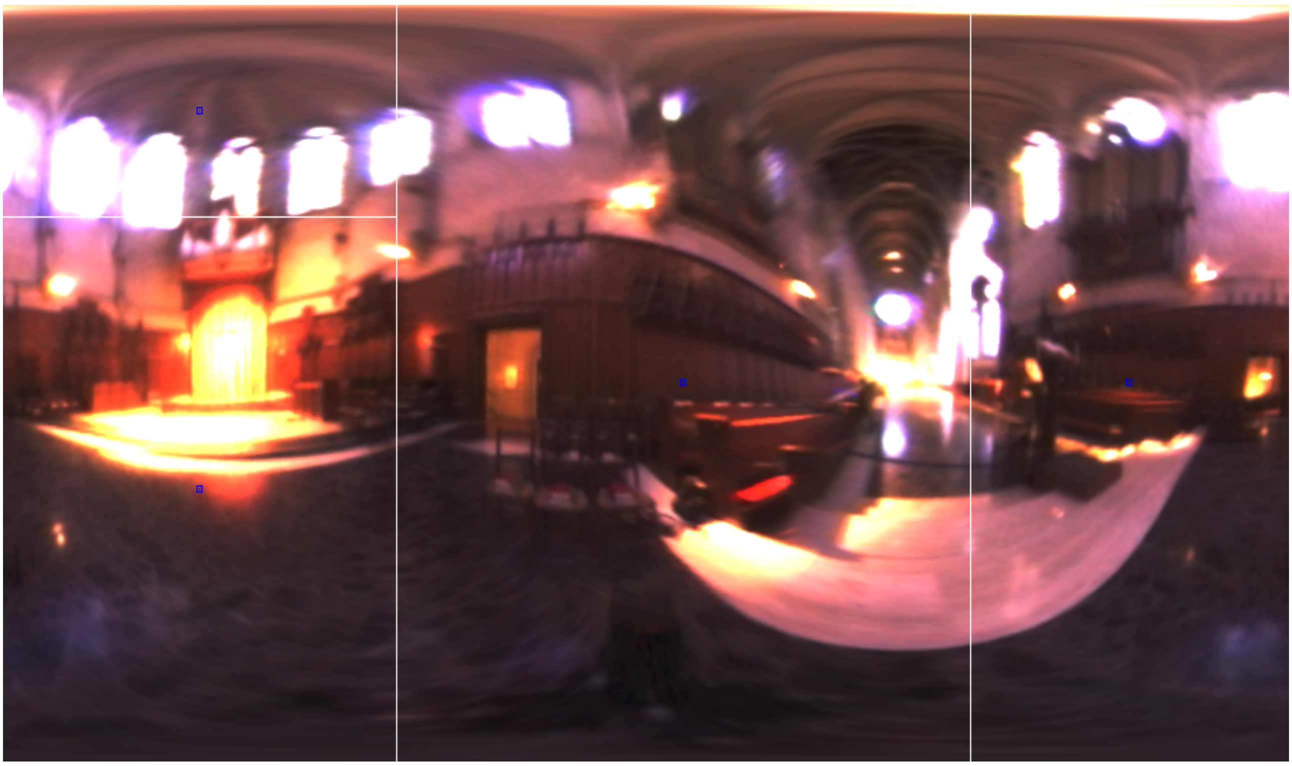


Figure 11: 4 partitions EM after gamma correction gamma=1.8 stop=3

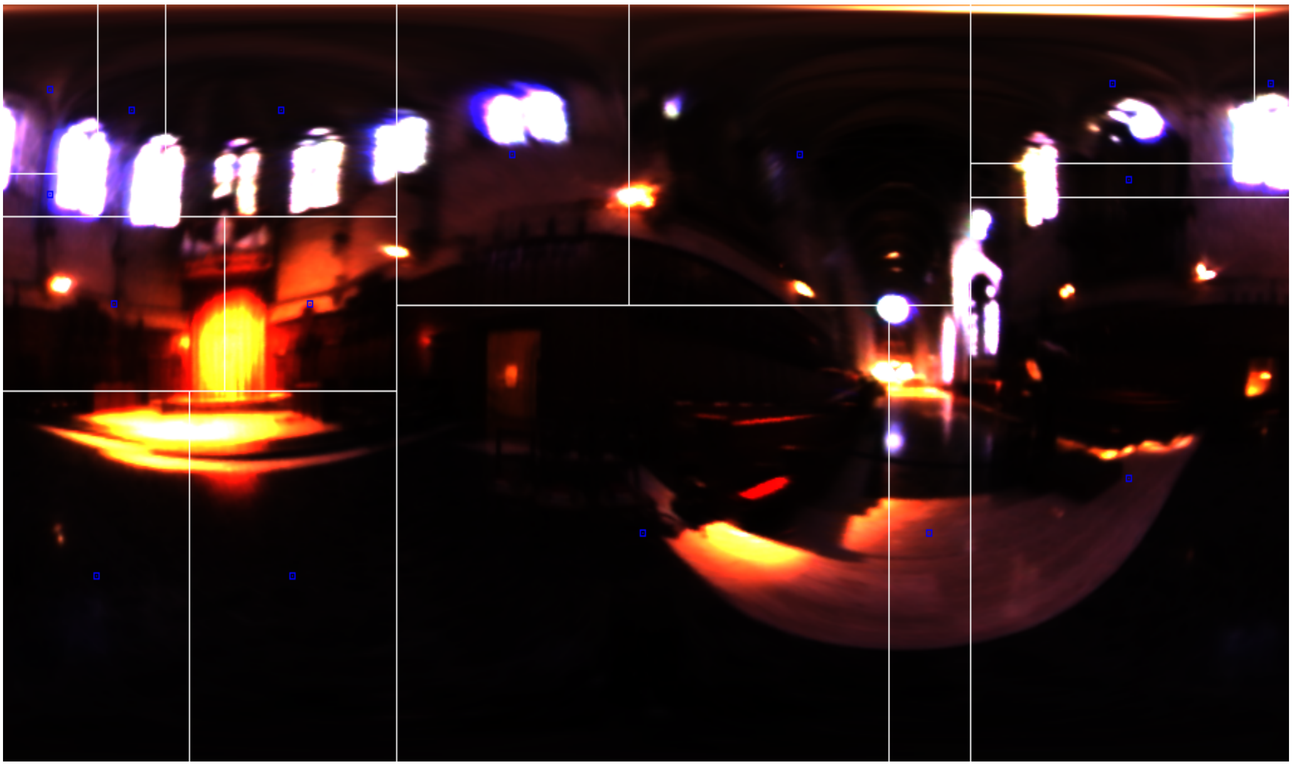


Figure 12: 16 partitions EM

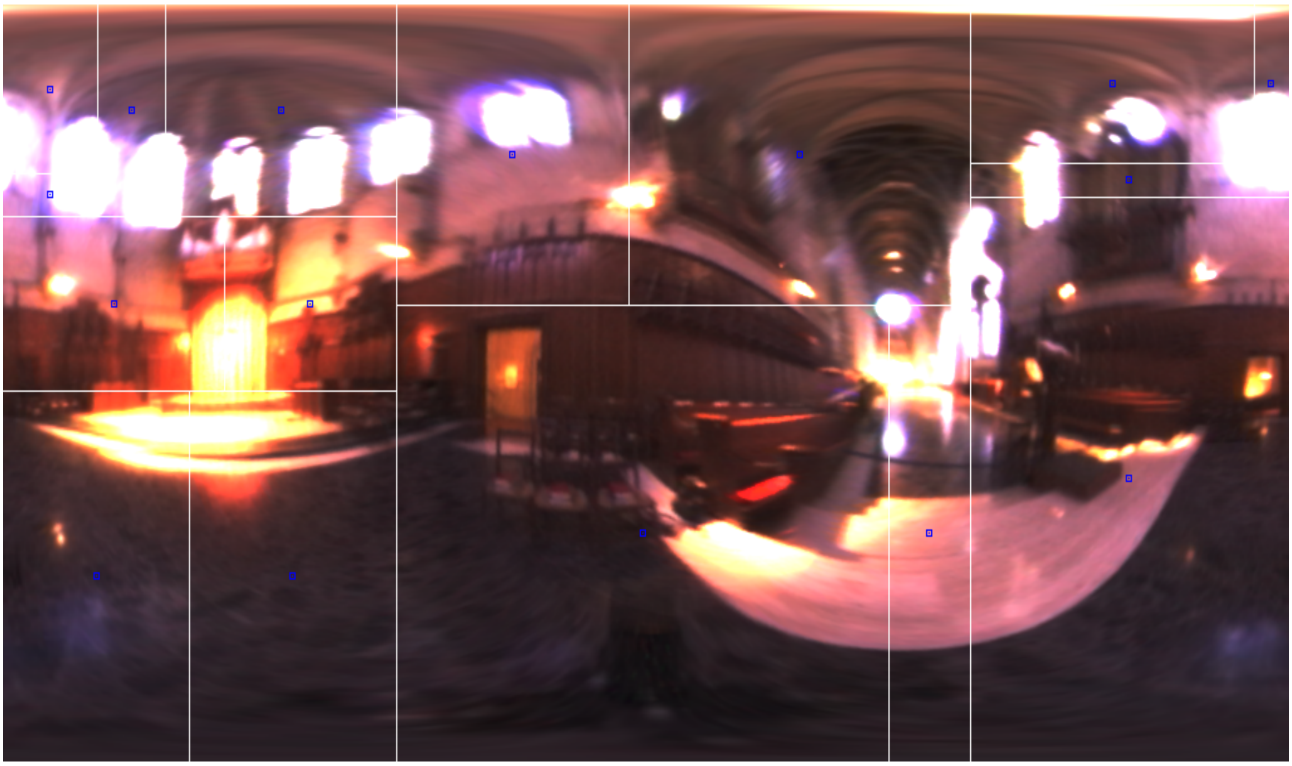


Figure 13: 16 partitions EM after gamma correction gamma=1.8 stop=3

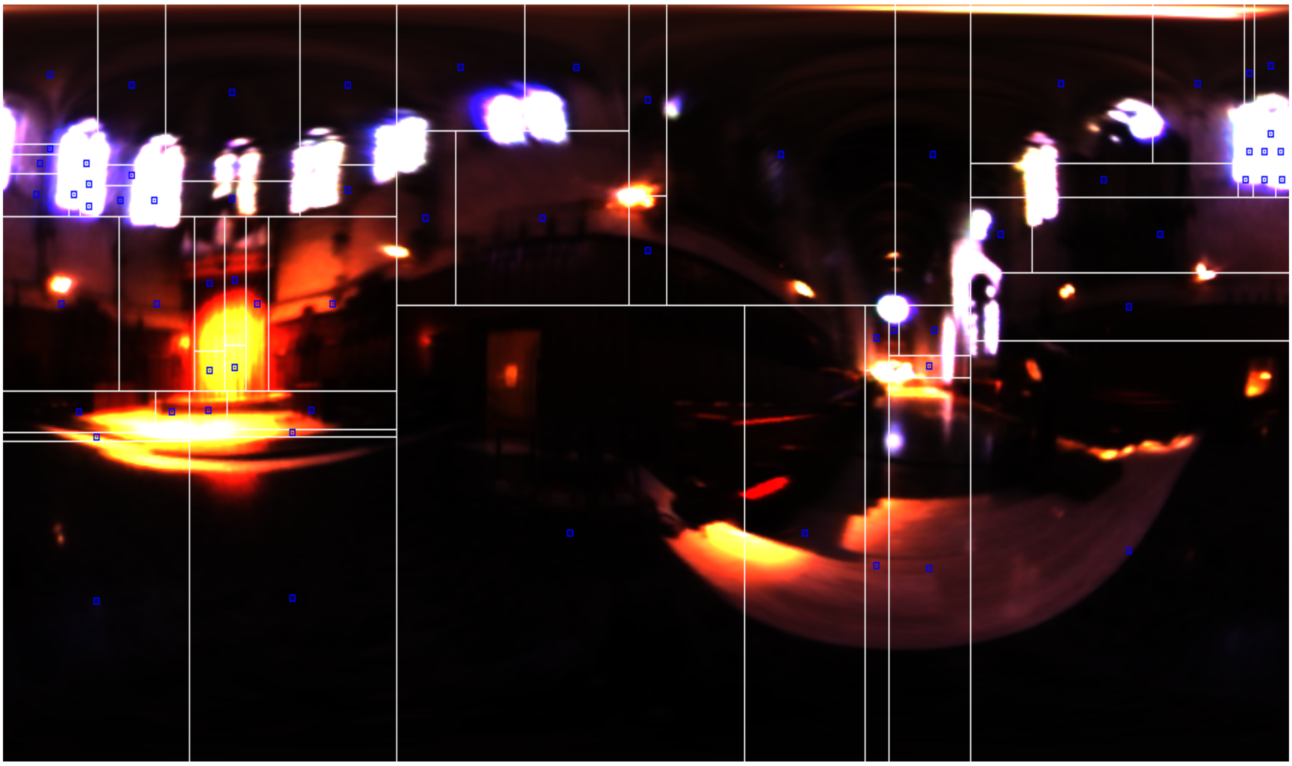


Figure 14: 64 partitions EM

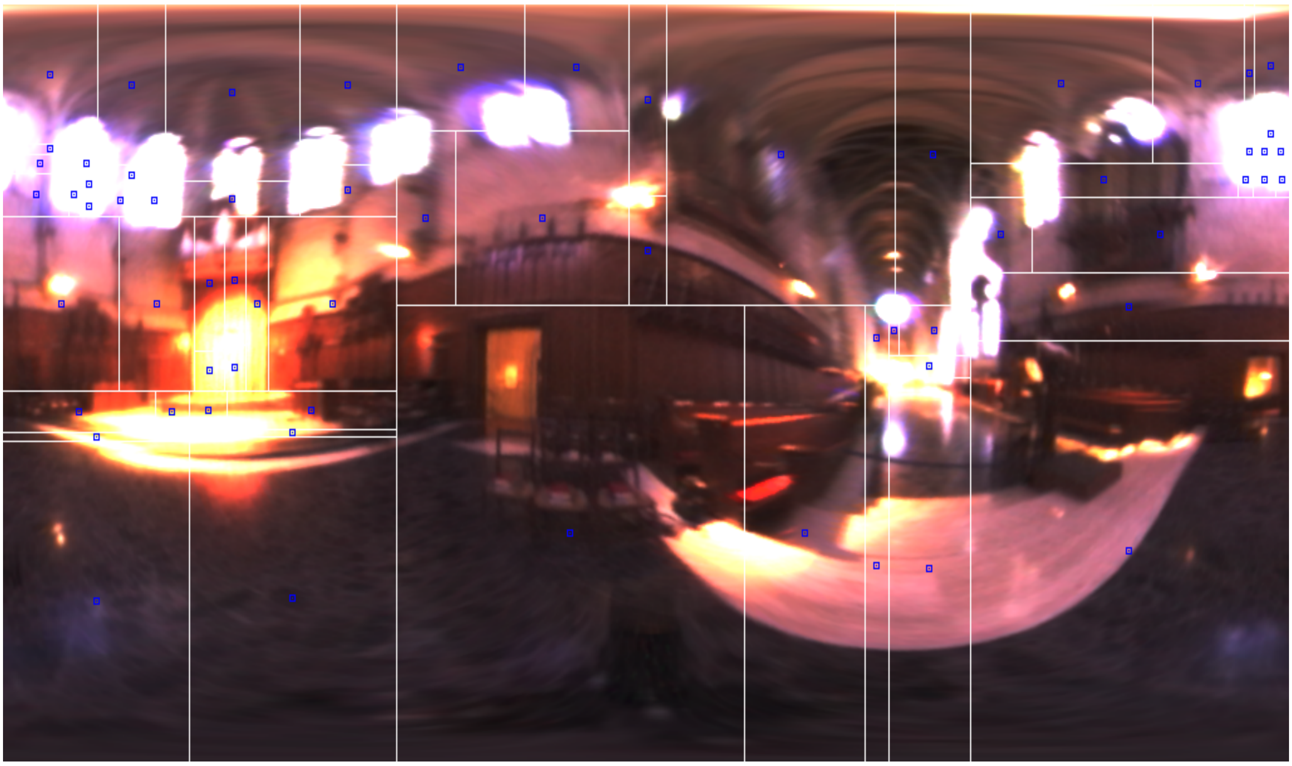


Figure 15: 64 partitions EM after gamma correction gamma=1.8 stop=3

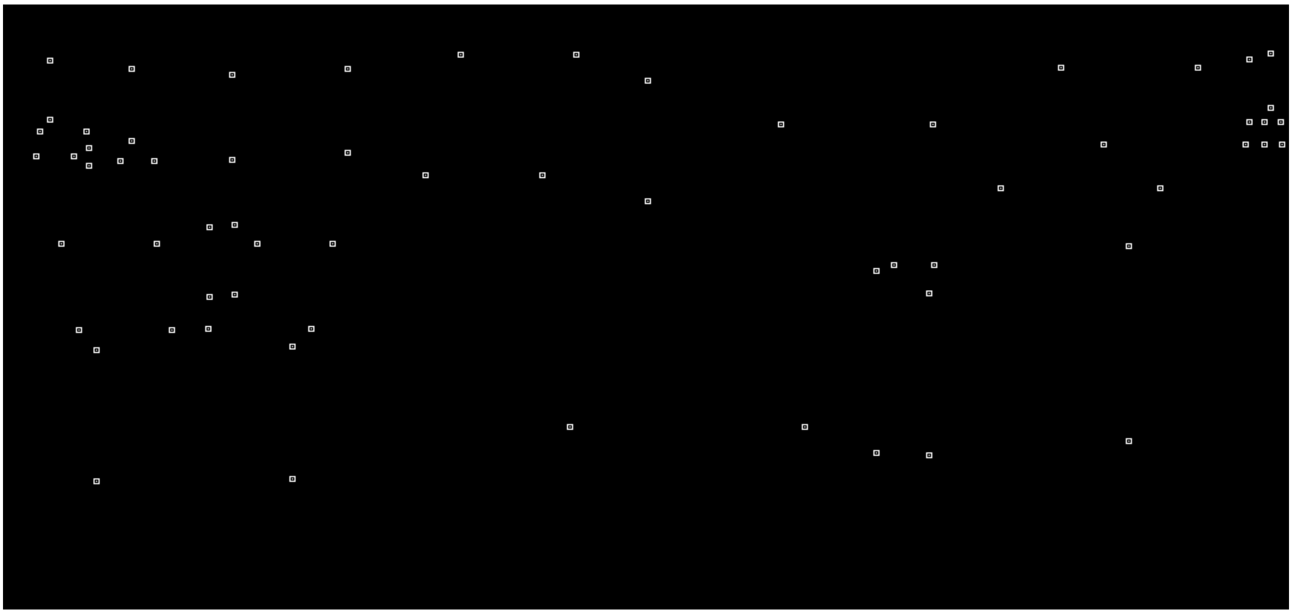


Figure 16: 64 partitions EM only the centroids total RGB before scaling

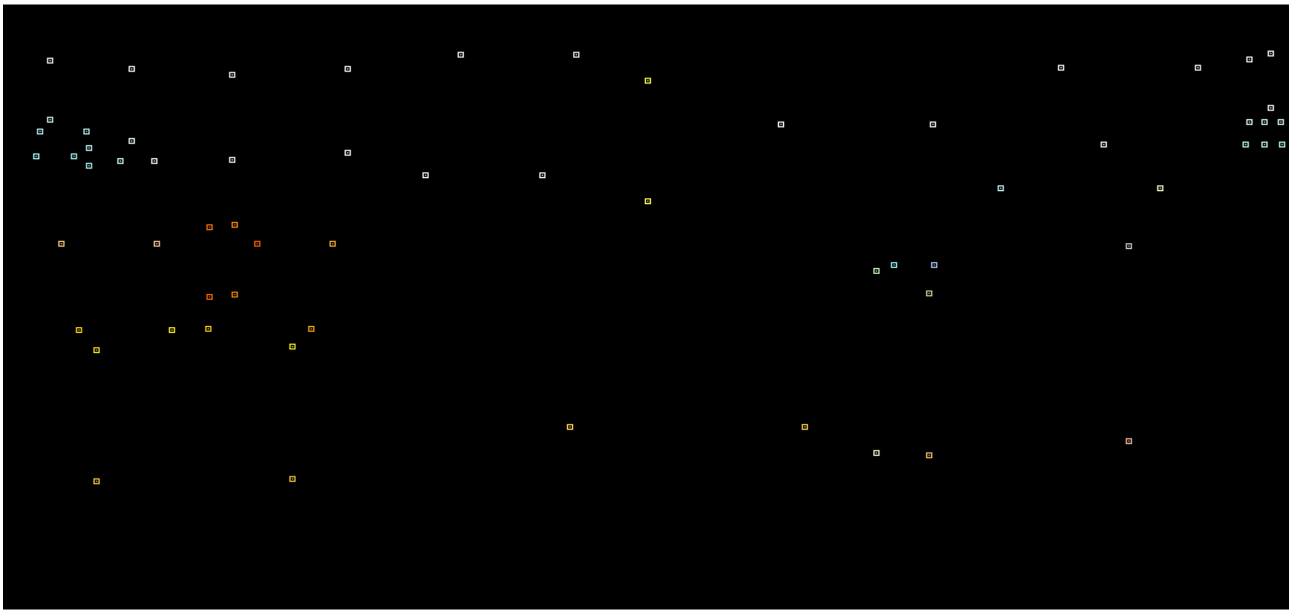
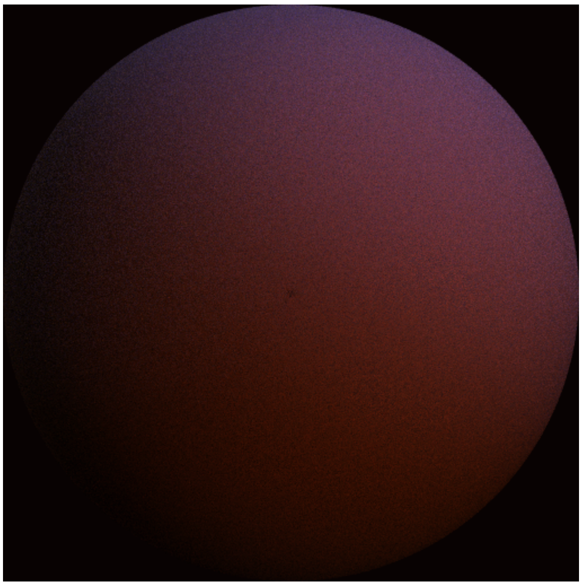


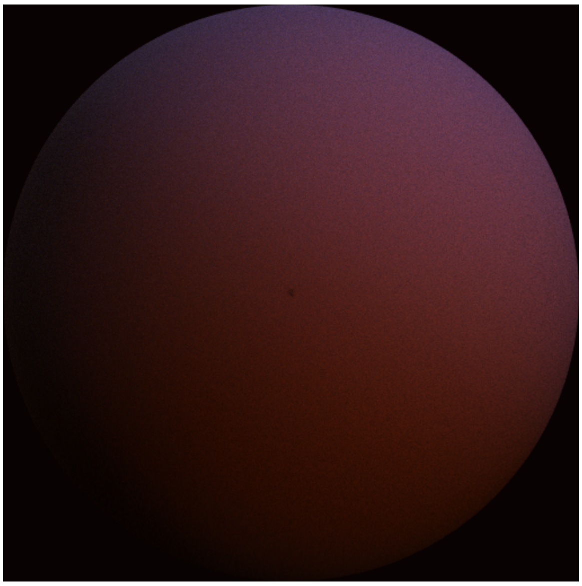
Figure 17: 64 partitions EM only the centroids total RGB after scaling

4 Render a sphere lit by Grace EM using PBRT

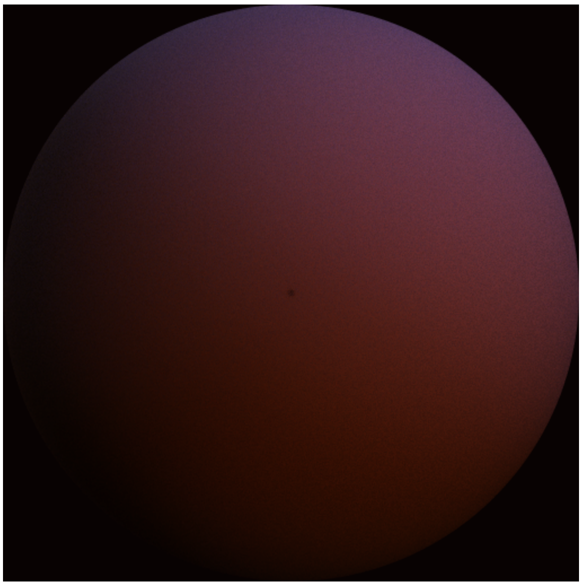
by change the sample per pixel in the `Simple_sphere.pbrt` file
under the `grace_latlong.exr` environment we can get the graph as follow as we increase the sample per pixel the generation time will be longer also there will be less noise. Also converting them into LDR format (.ppm) is shown after



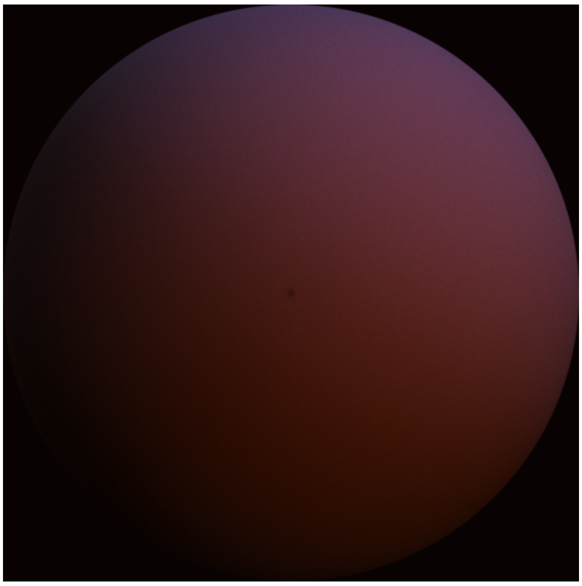
((a)) sample per pixel = 8



((b)) sample per pixel = 16

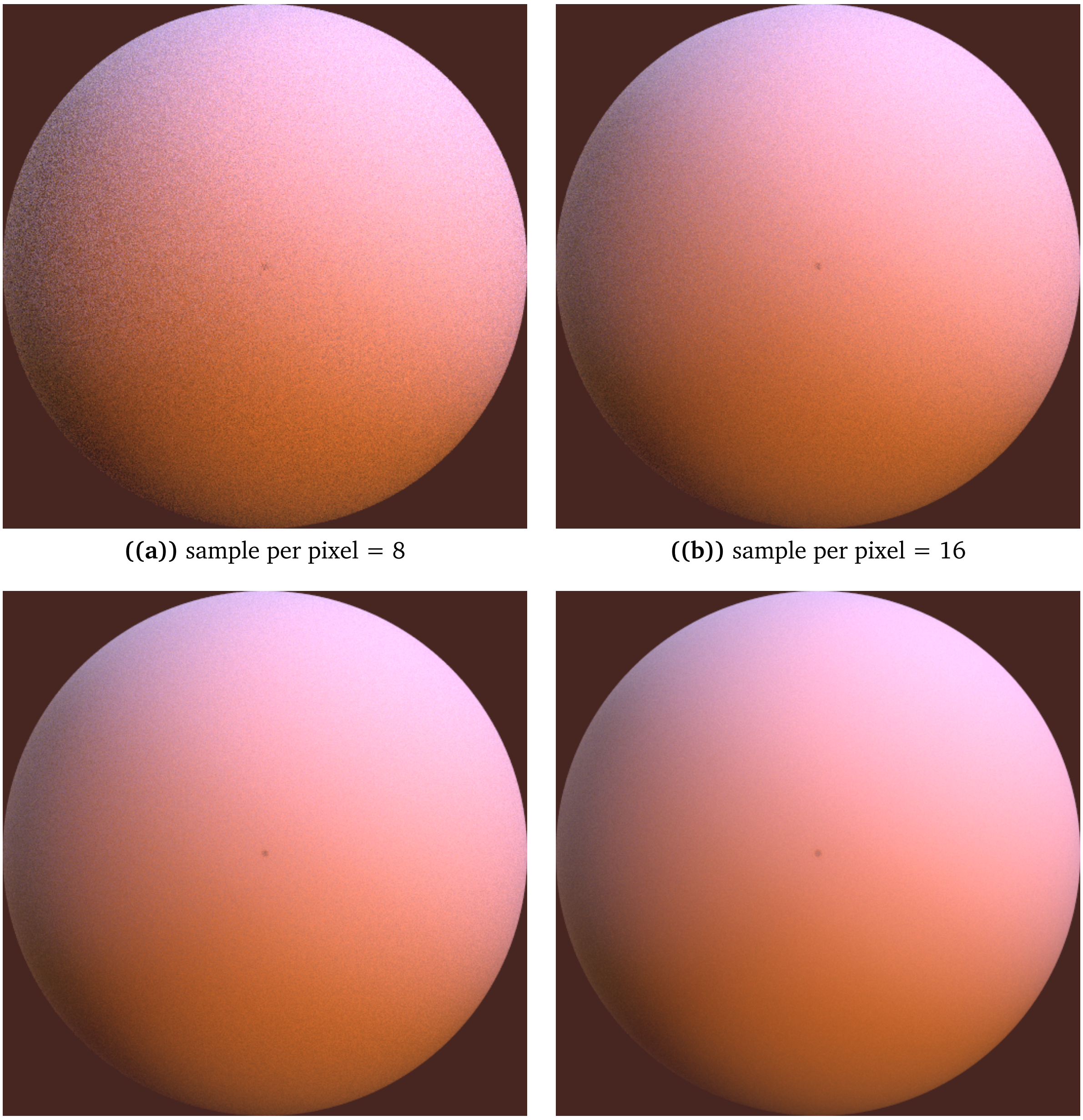


((c)) sample per pixel = 32



((d)) sample per pixel = 64

Figure 18: 2 x 2



((a)) sample per pixel = 8

((b)) sample per pixel = 16

((c)) sample per pixel = 32

((d)) sample per pixel = 64

Figure 19: 2 x 2